

### **In the Specification**

Please replace the paragraph on Page 1, lines 4 - 8 with the following marked-up replacement paragraph:

-- The present invention application is related to the following commonly-assigned inventions applications, which were filed concurrently herewith: U. S. Patent Application serial number 10/733,985, Patent \_\_\_\_\_ (serial number 10/\_\_\_\_\_); titled "Customized Subscription Builder" and U. S. Patent Application serial number 10/733,625, Patent \_\_\_\_\_ (serial number 10/\_\_\_\_\_), titled "Intelligent Subscription Builder". These co-pending applications are hereby incorporated herein by reference. --

Please replace the paragraph on Page 3, lines 3 - 6 with the following marked-up replacement paragraph:

-- The present invention provides novel techniques for programmatically assisting users in defining query parameters, thereby establishing ~~queries~~ more effective queries and/or queries with reduced errors. The queries thus defined may be used, for example, as the conditions under which the users selectively subscribe to data feeds. --

Please replace the paragraph on Page 9, lines 7 - 11 with the following marked-up replacement paragraph:

-- Referring now to Fig. 3, components and flows used in preferred embodiments of the present invention are illustrated. Query parameters 300, which may be programmatically determined or selected by a user as noted above, are input to a query builder component 310.

Current context information 330 may optionally comprise additional input to the query builder, and thus element 330 is depicted in Fig. 3 ~~[[used]]~~ using a dashed rectangle. --

Please replace the paragraph that begins on Page 13, line 5 and carries over to Page 14, line 2 with the following marked-up replacement paragraph:

-- In an aspect where the user has not made selections manually when the query-building process begins, Block 800 may comprise parsing the entire underlying content source. For example, when the content source is a Web page, the underlying content is typically specified in Hypertext Markup Language ("HTML") syntax. The parsing may therefore comprise identifying the tag names from elements of the content source. So, with reference to Fig. 1, by way of example, the column headings 120 - 160 might result from this parsing process. Block 810, in this aspect, preferably comprises consulting a database to determine whether some subset of those column headings are likely to be ~~[[or]]~~ of more interest to users than others (and the column headings, or a pointer thereto, are preferably passed to the database at Block 810). For example, while a user might be interested in searching on job title, the database might indicate that the "Description" column 140 should be omitted. (The description values might be perceived, for example, as being too "free form" to be useful when executing a query.) The consulted database may contain information that is based on gathered statistics, or it may contain mappings among terms, by way of illustration. As an example of the latter scenario, a mapping might be built as a set of terms, such as {"Location", "Job Title", "Education Requirements", "Salary"}, indicating that these four parameters represent the interests of a ~~"typical"~~ a "typical" user. The values returned from the database access of Block 810 might then be structured such that any additional

column headings (i.e., “Date Posted” and “Description”, in this example) have been pruned from the set that was passed to the database for evaluation. --

Please replace the paragraph on Page 14, lines 3 - 10 with the following marked-up replacement paragraph:

-- In another aspect, the parsed layout passed in Block [[800]] 810 may comprise a set of values which have been selected by the user. For example, the user might have selected (with a pointing device or other selection mechanism) the Location column 160 and Job Title column 130 from Web page 100 in Fig. 1. In this aspect, the database access may comprise retrieving candidate values to be presented in drop-down lists 422, 432, and/or candidate qualifiers to be presented in drop-down lists 421, 431. Furthermore, this programmatic selection of candidate values, and/or a programmatic selection of candidate qualifiers, may be used with the programmatic selection of query parameters just described (i.e., where the user did not select any columns). --

Please replace the paragraph that begins on Page 14, line 11 and carries over to Page 15, line 3 with the following marked-up replacement paragraph:

-- In one approach to programmatically selecting candidate values, a current context may be supplied to the query builder, as discussed above with reference to element 330 of Fig. 3. This current context may comprise an identification of the user, the user’s device, the user’s preferences, the user’s role, the current time of day, and/or a number of different types of information, as deemed useful for an implementation of the present invention. The manner in

which context information of this type can be retrieved is well known in the art. For example, if the current user's role is "Systems Administrator", this may be used as an index at the database to determine what values for "Job Title" are likely to be of interest for this user, and those values may then be used to populate the drop-down list 432. Or, if the user's current context indicates that he/she is physically located in the United States, then this information may be used to populate the drop-down list 422 with only job-postings locations in the United States. Many other examples of using context information will be obvious, once the teachings disclosed herein are known. --

Please replace the paragraph on Page 16, lines 9 - 15 with the following marked-up replacement paragraph:

-- In this aspect, the existing query parameters may have been determined programmatically (e.g., by selecting all column headings underlying a rendered Web page, as discussed above with reference to the job postings on Web page 100), or they may have been selected explicitly by a user. In this aspect, the query builder parses the existing-~~queries~~ query parameters (Block 900), and uses those parameters to access one or more databases (Block 910). The database access yields a set of one or more extensions, which are received at Block 920 and used to populate the query interface at Block 930. --

Please replace the paragraph on Page 17, lines 16 - 19 with the following marked-up replacement paragraph:

-- Referring again to Fig. 5, in this example, the extension returned at Block 920

comprises the “Salary” parameter which is automatically displayed for the user at 540. Preferably, a drop-down query qualifier list 541 is also rendered for each query parameter extension[[.]], along with a drop-down query value list 542. --